Technical Debt Calculation

Calculation of the Debt

Debt(in man days)	cost_to_fix_duplications + cost_to_fix_violations + cost_to_comment_public_API + cost_to_fix_uncovered_complexity + cost_to_bring_complexity_below_threshold + cost_to_cut_cycles_at_package_level

Where :

cost_to_fix_duplications	cost_to_fix_one_block * duplicated_blocks
cost_to_fix_violations	cost_to fix_one_violation * (violations - info_violations)
cost_to_comment_public_API	cost_to_comment_one_API * public_undocumented_api
cost_to_fix_uncovered_complexity	cost_to_cover_one_of_complexity * uncovered_complexity_by_tests (it is considered that 80% of coverage is the objective)
cost_to_bring_complexity_below_threshold	cost_to_split_a_method * (function_complexity_distribution >= 8) + cost_to_split_a_class * (class_complexity_distribution >= 60)
cost_to_cut_cycles_at_package_level	cost_to_cut_an_edge_between_two_files * package_edges_weight

All costs can be configured by going to Configuration -> Settings -> Technical Debt.

Default values for costs are :

What	Cost (in hour)
cost_to_fix_one_block	2
cost_to fix_one_violation	0.1
cost_to_comment_one_API	0.2
cost_to_cover_one_of_complexity	0.2
cost_to_split_a_method	0.5
cost_to_split_a_class	8
cost_to_cut_an_edge_between_two_files	4

This gives a debt in hour from which can be deducted the debt in man days and then in $$ by using values that can be configured :

What	Default Value
Hours per day	8
Price of a dev day	$ 500

Calculation of Debt ratio

The debt ratio is basically the debt divided by the Total Possible Debt (TPD) times 100.

The first thing we should calculate is the TPD on every axis :

Duplication
Given the number of blocks of duplication and the current %of lines duplicated, calculate the number of blocks there would be if it was 100%. If duplication is currently 0% then maximum number of blocks is calculated on the basis of 50 Lines per block. Total possible debt is the cost to fix those blocks.

Violations
Given the current RCI and the current number of violations, how many violations would be required to bring the RCI to zero. If RCI is 100%, the

number of violations used is NLOC/3. Total possible debt is the cost to fix those blocks.

Coverage
Total possible debt is the cost to bring coverage from 0% to 80%.

Complexity
Total possible debt is the cost to split every method and every class.

Comments
Total possible debt is the cost to comment every public API

Design
Given the existing edges between files, the cost of having to cut all of them (package tangle index = 100%)

TPD = sum TPD on each axis